

Learning Diary Thesis: Microservices and Internal Management Tools Developer in a Peer to Peer Lending Platform

Kbreab Hagos Tsegai



Author(s) Kbreab Hagos Tsegai	
Degree programme Business Information Technology (BITE)	
Report/thesis title Learning Diary Thesis: Microservices and Internal Management Tools Developer in a Peer to Peer Lending Platform	Number of pages and appendix pages 42 + 0
<p>Nowadays the terms frontend developer, backend developer and full stack developer are common among people who work in the information technology sector and can be noticed quite often in many recruiting websites. To new beginners, however they can be not as much as clear. Because with the advancement of web technologies, countless new frameworks, libraries and other tools are emerging and are causing “decision-fatigue” among new developers.</p> <p>The purpose of this diary thesis is to clarify the day to day experience and challenges of a backend services developer for a peer to peer lending platform. It shows the growth foot-steps of the developer after the joining the company with no prior experience as a backend developer. Thus, new developers and students interested in working as a backend developer can have an insight and better understanding of what it is to be a backend developer.</p> <p>It is conducted based on real assignments assigned by the Chief Operations Officer (COO) of the company. The task/s for the day, planned way of solving and desired results are detailed on a daily basis, then an overall summary on a weekly basis. It covers ten weeks of continuous work and shows that being a backend developer is not only about being able to write code, but also the importance of communication skills, attention to detail, teamwork and persistence.</p>	
Keywords Backend services, Peer to Peer Lending, SQL Server, Batch Operations, WCF Services	

Table of contents

1	Introduction	1
2	The start situation.....	3
2.1	Analysis of my current job situation	3
2.2	Interest groups in the workspace	6
2.3	Interaction skills in the workspace	7
3	Diary entries.....	9
3.1	Week 1 (19.02.2018 - 23.02.2018).....	9
3.2	Week 2 (26.02.2018 - 02.03.2018).....	12
3.3	Week 3 (05.03.2018 - 09.03.2018).....	15
3.4	Week 4 (12.03.2018 - 16.03.2018).....	15
3.5	Week 5 (19.03.2018 - 23.03.2018).....	18
3.6	Week 6 (26.03.2018 - 30.03.2018).....	22
3.7	Week 7 (02.04.2018 - 06.04.2018).....	25
3.8	Week 8 (09.04.2018 - 13.04.2018).....	28
3.9	Week 9 (16.04.2018 - 20.04.2018).....	31
3.10	Week 10 (23.04.2018 - 27.04.2018).....	34
3.11	Week 11 (30.04.2018 - 04.05.2018).....	37
4	Discussion.....	40
	References	42

1 Introduction

Fellow Finance Oy is a lending platform established in 2013 that provides various loan offers for both consumers and Small to Medium Sized Enterprises (SME). Even though Fellow Finance is a young company, it has grown to become the biggest lending platform in Northern Europe. The reason of being the first crowdfunding platform that serves both consumers and businesses has contributed to the growth of the company and its customer base. As of 2017, Fellow Finance has over 280,000 customers from 43 different countries. Currently it has over 20 employees with offices located in the cities of Helsinki and Turku, Finland.

The existing online platform of Fellow Finance is developed and maintained in house by a team of highly skilled and experienced software engineers. These engineers have many years of experience in automating banking processes and building the IT infrastructure of several Finnish banks. I joined this IT development team in February of 2017 as a backend C Sharp developer to help contribute in the process of creating and upgrading different financial microservices that run the business processes of Fellow Finance.

In the first two weeks, I was introduced to the customer service and software development team's operations. This helped me to get first insight on the working culture and practices of the company. Important to mention is, Fellow Finance has a flat organizational structure, that has enabled staff members and executives to communicate directly without the need of middle management, which has resulted in faster decision making, productivity and growth of the company. Following this, I got briefly introduced to the software development team and its various kinds of backend services that provide support to consumers and customer service through web and API interfaces.

To work as a backend developer for Fellow Finance, one is required to have not only technical skills, but also other skills such as attention to detail, proper communication ability with different teams of the company, flexibility to work both independently as well as within a team and logical thinking abilities are some of them. The famous quote by *Lee Iacocca*, former CEO of Chrysler Corp, says *"You can have brilliant ideas, but if you can't get them across, they won't get you anywhere"* to clarify the importance of communication skills for achieving positive results.

The basic technical requirements for working as a backend developer in Fellow Finance are having a comprehensive understanding of Object Oriented Programming principles and proficiency in using the C Sharp programming for writing code. In addition, the know-

how to visualize program flow, understand and translate pseudocode to a clean C Sharp code and testing.

As Fellow Finance is a data driven platform, knowledge of the concept of databases in general and the ability to manage SQL Server databases is required in the day to day tasks. Some of the database operations in Fellow Finance include: creating of tables and views, writing different kinds of queries to manage data and periodic performance optimization operations.

Although not required and can be learned during the process, some knowledge or experience of Azure cloud services, Windows Server environment and Customer Relationship Management (CRM) tools are a plus and play a big role in speeding up the development process.

2 The start situation

Following is the details of what I do basically, types of tasks I am responsible for and the experience I gained with in the year I worked for Fellow Finance.

2.1 Analysis of my current job situation

It has already been one year since I started as a backend developer in Fellow Finance, and currently I have become one of the senior developers of the company. Although I have been participating in different parts of the whole platform, my main list of assignments includes:

- Developing and updating of microservices that provide support for our customer service CRM, web platform and external business partners
- Creating API's for our web platform and external business partners
- Developing new and updating existing business processes
- Database management
- Creating different types of business report layouts
- Developing report generations services
- Developing document generation services
- Batch operations development and updating existing ones
- Office 365 administration
- Technical IT support for customer service

The nature of the above-mentioned list of assignments vary greatly in the purposes they serve; thus accordingly, their implementation also differs from one another. One common thing that can be noticed amongst the programming assignments is that they involve data access logic at some points of their implementation. For the sake of clarity and being concise for the introduction, I will take the liberty of using a generic way of explanation in describing the steps for completing the assignments.

For assignments related to service designing, I first start implementing them by identifying the service type and the data it returns. To mention some of our service types: we have services that return data from database, services that alter data on database and services that make logical decisions and relay data to processes and other services. In addition to this, for services that interact with database, I also make sure that the target database entity has the right fields required by the service. This is an important step, because new services sometimes require for fields that does not exist. In such cases, I usually alter the database entity by adding those new fields and then make sure the new changes are reflected to locations where the entity is used. After completing the above-mentioned steps, follows the writing of the business logic of the service. In this step, attention to detail is

very crucial. Missing a requirement or misinterpreting one, can lead to unwanted results which causes loss of time and resources.

The logical implementation of business processes is also similar to creating services, except that processes are comprised of smaller steps of execution where they are dependent on one another and the queue of their execution matters. When creating a process, I identify and define the steps that make up process. Then, these steps are queued and implemented in a way to achieve the desired result of the process.

Regardless the type of assignment, after the initial completion of the business logic, I perform complete testing to ensure they are behaving as intended and producing the expected results. It is after successful and thorough testing that I push the code to master repository and publish the new changes to production server.

In completing majority of the assignments assigned to me, a comprehensive knowledge of c-sharp programming language and object-oriented programming is needed in terms of technical aspects. In addition, some tasks also require working experience of SQL server database management, xml and XAML. But technical knowledge alone is not enough to get a job done. The know-how of creating the logical structure and algorithm for the solution of the assignment at an abstract level is key in the initial stage of implementing the assignment.

The above listed technical skills give an insight of what is expected of someone to work as a backend developer, and the mention of services and processes, show a sample of the variety of the types of projects that exist on the server side of Fellow Finance. To complete assignments successfully, one needs to understand what he/she must do. In my case, I always use a requirements document produced by the head of our IT development team, to be able to understand the details of an assignment and its desired goal. A requirement document usually contains a detailed and easy to understand description of what needs to be done, the goal of the assignment, the allocated time frame and sometimes mentions of external collaborators if any.

Within the one year of my employment in Fellow Finance, I was able to expand the frontiers of my knowledge and experience in technical, business, financial and communication aspects. The flat organizational structure of the company and the relatively few number of developers in our software development team gave me the chance to take the responsibility of managing and developing new features for our Customer Relationship Management tool. As a result, I was able to learn and develop several new features for our CRM. To

mention some, I created a collection calls dashboard that tracks customers with late payments, offers them a new payment plan for the late payment by adjusting the date, installment amount and other factors, and finally sends them an automated email or SMS based on the type of payment plan they preferred. Also, in December of 2017, I implemented a project that generates invoices on a daily basis using batch operation and sends those reports for Polish customers using a SOAP protocol. The challenging part was that the reports had to be signed before sending, using a digital certificate provided by the mail delivery company located in Poland, and then the customer details and address window of these reports had to be adjusted to a specific location with embedded fonts that include diacritic Polish letters. Some of the knowhows I got as a result of being tasked with such assignments are:

- Expand my knowledge and experience of the C-Sharp programming language
- Learn about SOAP protocol specification and sign SOAP envelopes using binary digital certificates
- Create dynamic reports with embedded fonts
- File processing and management
- Create web components for CRM using XAML and C-Sharp
- Properly utilize online help sites such as stack overflow

In the first three months after I started working in Fellow Finance, some of the assignments that involved creating batch jobs, dynamic report generation, file and directory management were challenging, as I had no or only limited prior experience related to these kinds of assignments. For instance, I had an assignment to create a batch job that must run each morning on weekdays to deliver invoices to our customers. In this task the challenge was integrating this batch job with our CRM so that it can be easier for our customer service team to follow its status and configure the time it must run from an interface in the CRM. Another challenge was to modify the generation of invoices for Polish customers by embedding fonts inside the generated pdfs, sign them with binary digital certificate and send them to a mail delivery company using a SOAP envelope. The reason for embedding fonts, was that the Polish language has additional letters called diacritics. These additional letters had to be embedded during report generation, so that the remote SOAP server of our mail delivery partner company can understand the delivery address of our customers correctly.

Being tasked with assignments like the above-mentioned ones and many others with similar or more complexity, I had to face a lot of challenges in the process of solving them. Some of the challenges were: lack of knowledge and experience for some tasks, shortage of documentation and clarity problems in the requirements document. Nevertheless, for most of the assignments I was able to complete them efficiently with in their allocated

time, through doing my own research and getting help from online developer's community and my seniors in the team. This in turn contributed to a significant growth of my knowledge and experience in software development.

Currently, I am at a stage where I am able to understand assignment requirements without difficulty and develop several kinds of projects that touch different paradigms of programming. Most of my work is focused towards developing services and new features for the CRM. I aid the customer services teams with technical matters, accept bug reports and take action on them. Since I joined Fellow Finance, through the combined effort of everyone, sales were more than doubled, we opened the German market, and now we are getting ready to open a new one for Swedish customers.

To stay relevant in my field and expand my knowledge, I have plans to invest my time for the future in studying the newly released ASP.Net core 2 framework. According to Microsoft's official website, ASP.Net core is a complete recreation of the previous ASP.Net framework. To get an insight, I took some introductory courses of ASP.Net core. It follows a minimalistic approach, and one can add the "things" needed during development. In addition, learning the frontend framework Angular 5 is also in my to-do list. It integrates well with ASP.Net core and enables for the creation of UI components that are reusable.

2.2 Interest groups in the workspace

Fellow Finance is a relatively small company having a little over 20 employees. There are few internal groups within the company, but being the largest peer to peer platform in Northern Europe, it has many external interest groups. Following is the list of these groups:

Internal groups:

- Sales team
- Marketing team
- Customer service team
- Finance team
- Administration

External Interest groups:

- Loan Insurance partners
- Credit Bureau partners
- Loan Brokers
- Mail delivery companies
- Customers

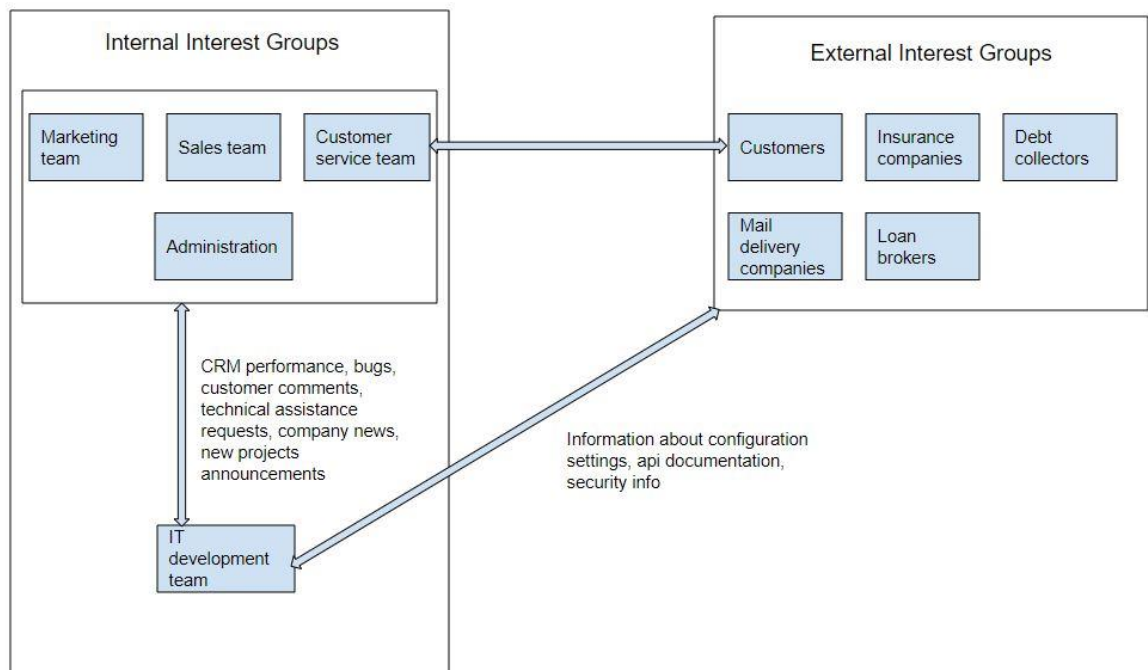


Figure 1. Interest groups and their interaction context towards IT development team

From the internal interest groups of our company, I receive different kinds of information, but the most relevant ones are those related to our Customer Relationship Management (CRM) and customer comments received by our customer service about the web platform. These information and interactions are about the performance of CRM and web platform, list of errors found, requirement for technical assistance, suggestions for improvement and new ideas. They help me notice unseen loopholes, fix unhandled errors and improve the performance of our CRM. With external interest groups business logic is conducted through web interface, web Application Programming Interface (API) and file transfer protocols like Secure File Transfer Protocol (SFTP). We exchange information about settings, API documentation, document format and security, which is essential to successfully setup and get the business start working.

2.3 Interaction skills in the workspace

In Fellow Finance English is used as the medium of communication. Within my team we usually conduct face to face short stand up meetings to discuss about ongoing work and future plans. Outlook is used for sharing requirement documents and other information. In addition, skype for business is used by each employee for instant messaging service within the company and to call external interest groups.

Some challenges happen when the interactions with the non-IT teams are about technical matters. Because at times it is harder for them to explain it in a way easier to understand

in technical matters. In such cases, if I am working remotely, text-based communications are not enough and I often use remote access tools like TeamViewer to better understand the situation. Otherwise, most of the time I work from office and we solve any interaction skills with face to face meetings. If the case is beyond my ability our seniors take part in providing a solution.

3 Diary entries

3.1 Week 1 (19.02.2018 - 23.02.2018)

Monday

As I mentioned in the introduction part, we are opening a new market for Swedish customers. Whenever we enter a new market or create a new partnership, it is often that we make changes in our business logic so as to be able to provide support for the customers of the new demographics. Today, my tasks were to create custom data validators for Swedish customers. These validators are for validating a customer's Social Security Number(SSN), bank account number and phone number. The reason is that the format of these values adheres to specific format that follows the Swedish standard and law which is usually different from other countries specification. Within the values of the SSN and bank account numbers there is a check digit character, usually a number, which denotes the validity of the given value. This value in case of SSN is generated through some mathematical computation of the other digits, and in case of bank account numbers it is specified according to the country's standard for bank account numbers.

After reading and understanding the requirement and structure for each, I used a pseudo code to help me draft the logic. Then I implemented the code using typescript for front end and C Sharp for server validation. I tested them using different test cases, and handled exception. After confirming their output was as expected, I committed the code and published to production. In this task, I used Regular Expression (Regex) to check that the format of the values conforms to their defined standards. In doing so, I learned more about Regex and how can I utilize it to perform more pattern matching.

Tuesday

Today's task was to create a batch job that sends data in an Extensible Markup Language (XML) format to a new Credit Bureau Partner from Sweden. The XML file is required to have a specific structure defined in the documentation we received from the new partner. In order, to fetch data and serialize to this specific format, first I created a class having a corresponding structure with all the required properties and XML data annotations. Then, I added the namespaces and serialized it to an XML doc using C Sharp's built in serialize method.

One particular requirement for the XML to be generated was that, elements with no data must not be serialized as self-closing tags. The XML serializer serializes elements with empty data as self-closing tags, which I had to override this behavior to have elements with a separated closing tag. If the method `WriteFullEndElement` of the class `XmlTextWriter` is used when generating an XML document, empty elements will be generated with full end closing tags (Niel M. Bornstein, 49). I used the above specified method in the book, to override self-closing tags and generate empty element with full end closing tags.

Wednesday

My objectives for today are make changes to one of our services for adjusting loan applications and prepare a user profile, CRM access and office 365 for a new employee to work in the customer service for Swedish customers.

The task for updating the adjust loan applications service is that, new parameters for increase amount and opening fee must be added so that our customer service can be able to manually adjust the amount our customers want their credit applications to be adjusted to and insert the respective fee.

After modifying the service, I tested it and the changes were published to our test server. Then I updated the service references from our CRM get the latest version of our services and do testing from the CRM also. I updated the adjust loan applications dialog by adding new input controls to that bind to the newly added parameters to the service. Finally, I tested the new changes and published both the services and CRM to production servers.

For our new employee, I created a user profile in our server active directory, setup his new laptop with the necessary office 365 installation and our company's VPN so that he can be able to work remotely sometimes.

There was not particularly anything new about today's tasks, that I got to learn new things from. I completed them in few hours and notified our customer service of the new changes to the adjust credit applications process.

Thursday

Today I got a notification some of our batch operations, that send reminder, warning, invoice and termination letters to our polish customers through a Polish mail delivery company, failed to deliver the letters. Our batch jobs interact with the Polish mail delivery company through a Simple Object Access Protocol (SOAP) interface to send and receive letters and other kinds of information, such as report on the number of letters we sent, successful and failed deliveries and cost.

After reviewing the log files from our batch operations server, I found out, it was internal error of our mail delivery partner server. I notified them of the situation and had to wait until they fix the problem with their server, so that I can run those failed batch jobs again to send the various types of letters for delivery.

Friday

For this day, the objective is to create a new cold calls management dashboard and add it to our CRM. The objective of this new dashboard is to list customers selected for cold calling and track status of the calls. Our marketing team can then utilize this dashboard features for further marketing and advertising purposes.

I started doing this task by first implementing the backend services necessary for this dashboard. These services are for fetching customers list, for updating status and for sending email or SMS based on customer preference. After testing these services from server side, I published them to test server to complete the front-end task of creating the dashboard user interface. I used Extensible Application Markup Language, or Extensible Application Markup Language (XAML) to design the layout of the dashboard and its components. Then I added the dashboard data properties using XML.

I tested that the dashboards listing is working correctly and also the functionalities for updating call status and sending messages. New changes were pushed to both services and CRM production servers.

Weekly analysis

This week was one of the preparatory weeks for us to enter the Swedish market. I have been doing small fixes and updates to existing code base, and creating ones such as data validators and batch jobs as part of us getting ready towards Swedish consumers.

When creating the validators for Social Security Number(SSN) and Bank Account Number, one of the first steps is to check that they follow the format based on the issuing country or banks policy. For such cases regular expressions are one of the most recommended tools to use. According to Joe Booth, “a regular expression, is a pattern of characters that are used to find matches in a larger string or text” (Regular Expressions Succinctly, 2017). This book helped me learn a lot about pattern matching, important keywords and shortcuts. As a result, I was able to get a more deeper understanding of regular expressions and was able to create expressions that can do complex pattern matching of bank account numbers and social security numbers of different banks and countries.

Regarding data transfer, my experience was mostly with using JavaScript Object Notation (JSON) for such tasks. In one of this week’s task related to generating XML file by serializing C Sharp classes, I learned about data serialization to XML, adding namespaces and many other useful information from Microsoft’s online documentation about XML, existing solutions on Stack Overflow and also from the comprehensive book about C Sharp by *Andrew Troelsen and Philip Japikse, C# 6.0 and the .NET 4.6 Framework, 2015.*

3.2 Week 2 (26.02.2018 - 02.03.2018)

Monday

On this day there was a major service disruption with the internet and virtual private network (VPN) service provider for our company. When I arrived at the office first the VPN service was not working properly which was followed by the interruption of the internet service. Then after an hour since the interruptions, both the VPN and the internet went off completely and the service provider company made a press release regarding the disruptions. The first step I took was set up a 4G access point so that our customer service can keep on serving customers through skype and email. As a result of the service disruption, the configuration of our local office network was also affected causing the CRM not to respond to http requests for a particular URL. As a solution to this, I used an alternative URL which accesses the CRM from our VPN located in Azure and tested the CRM. The test was successful, and information was sent to everyone to use this particular URL until the problem with our internet provider gets solved.

Moreover, the configuration of our office router was also corrupted. I tested it by sending test packets to the nodes of our network and the reply revealed there was a network error.

Tuesday

Today, we are expecting one technical guy from the internet provider of our company to fix the corrupted network configuration and local VPN connection. I will also go to the electronics shop Verkkokauppa, to make purchases for 24-inch display, headsets and some USB hubs.

Wednesday

This morning our customer service for Finnish and German markets received several emails and phone calls from our customers, regarding them receiving the same invoice by email up to three times on the same day. This was a very important information and was given highest priority to be solved as soon as possible.

Invoices are sent to our customers by mail and email periodically using batch job operations that run daily, based on their scheduled time. Our invoice delivery batch operations were following the steps of delivering invoices first by email then by mail. Mails are sent to remote file servers of partner mail delivery companies by using data transfer protocols such as SFTP (Simple File Transfer Protocol) and SOAP (Simple Object Access Protocol).

The problem was found to be that, although email delivery was successful, but mail delivery was not always successful. The reason was that the file server of one of our mail delivery partner companies had some internal errors which caused our mail delivery batch operations to fail. And when failed our batch operations were trying to send again both the already delivered emails and the failed to be sent mails. This caused some of our customers to receive the same email more than once.

I fixed this bug by reconfiguring our batch operations to send emails only after invoice files are sent successfully to the remote servers of our partner mail delivery company. I also updated the log information to be more detailed, which could help us track future errors and bugs more easily.

Thursday

As a continuation towards getting ready to support Swedish customers, we have got an email informing about a pre-approval loan contract word template is ready, for which we have to write the logic to fill the dynamic values of the document and then generate a pdf document of the loan contract to be sent to a customer.

Although majority of the dynamic fields in the loan contract templates for the various countries we serve are the same, there are some specific ones that are often based on the target country's language and credit legislation. In this task, there were some fields specific to Swedish loan contract for which I had to modify the existing loan contract generation method to support these new fields.

After the changes to the document generation method, I made several tests to check document is being generated with the correct data and intended layout after the dynamic fields are filled. An important lesson I learned from this task is how the representation of some generic methods in a loosely coupled manner can help in scaling up an application by only making small changes.

Friday

This morning I received a high priority task related to our loan cancellation process. Our business seniors made some new decisions which required some new steps to be added to the loan cancellation process. The task is to check for the target loan that there it does not have loan allocations with a specific status.

It is important to note that, while creating a new one or modify an existing process by adding or removing a process step, there order of the steps matter and can affect the output of the process. The order of these steps is usually determined by the purpose of the process.

After confirming that the position of this new step is to be the first step of the process, I added this new step to the process and wrote the code which check for the status of the loan allocations for the target loan. I used lambda expression to traverse through the various loan allocations and check if their status is the same as the particular status to be checked against.

For allocations that satisfy the check case, I added logic to log the identity of the first allocation that has the same status as the check status and interrupt the loan cancellation process. Logging of computation information and errors plays an important role in tracking application status and debugging errors.

Weekly analysis

I have spent majority of this week dealing with the internet problems and network configuration corruption caused by some technical problems that occurred to our internet service provider. As an initial step, I used the ping command to check the network status. Jeffrey S. Beasley and Piyasat Nilkaew, define the ping command as a way to verify that the devices in the network are communicating properly (2012, 44). A technical guy was sent by the service provider to fix the problems, and while doing so, I had spent some time configuring temporary hotspot access point for each customer service personnel so that they can continue to serve customers.

In addition to helping customers with setting up temporary internet connections, I also made several code changes and bug fixes related to batch operations that send different kind of invoices to customers.

This week I learned how crucial it is, to identify and act upon emergency situations. The internet service disruption was one case where after identify the problem, I was able to quickly setup temporary internet hotspots for our office until the problem got solved.

3.3 Week 3 (05.03.2018 - 09.03.2018)

Winter break vacation.

3.4 Week 4 (12.03.2018 - 16.03.2018)

Monday

One of the tasks I did on Tuesday of week eight, was creating a batch job that sends data in an xml format to a new credit bureau partner. After I completed that task, we sent a sample of the generate xml data was sent to the credit bureau company to check the content and format structure. The receiving end analyzed the data we sent and confirmed it includes the required data and is formatted according to their specification.

We received the confirmation this morning, and thus today the plan is to finalize the data transfer service and make it ready for sending production data. The first step was acquiring configuration settings for their production file server. After I got this information I updated this partner's configuration information in our config files. For this new config changes to take effect I rebuilt the project that contains our config files and tested once again to confirm the connection to their production server is successful. The test was a quick test just to make sure the new config settings to their production server took effect successfully. I added exception handling and logging, then tested from debug and monitored the steps. After making sure everything was working as expected, I committed the changes and published it to our server.

Tuesday

It is a known fact that for a business to grow, keeping customers happy and satisfied is one of the most sought-after business processes. Without a happy customer it becomes hard to maintain a sustainable business growth.

Thus, today's task is related to customer satisfaction and it is to create three new SMS message templates and implement the SMS sending C-Sharp code for our Polish customers. After learning that some of our Polish customers prefer for some types of information to be sent to them by SMS rather than by email, we decided to implement them as soon as possible and as a first step, our Polish customer service team created the SMS versions of these information.

I received the word document containing this information and created the corresponding SMS templates on our CRM. Following, I linked these templates with our Poland brand entity and finalized it by adding the code for sending SMS messages.

Wednesday

We have two new employees starting on Monday 19th of March, 2018. Today, I have received two brand new laptops and my plan is to prepare them for the to be starting employees by installing the necessary programs, setting security credentials and adding them to our Virtual Private Network (VPN). In addition, I have to purchase Office 365 premium business license and set it up for them.

First, I installed the programs Adobe pdf reader and the Firefox browser that provides support for Silverlight plugins. Some interfaces of our CRM are built using the Silverlight

framework and currently they work with the browsers, internet explorer and a specific version of Firefox. Although most of the time we use internet explorer, we also install the above-mentioned Firefox version as a backup in case internet explorer encounter some problems.

I then installed the program and certificates for our VPN, created user accounts in our active directory with security roles based on the teams they are going to work and CRM access accounts.

Thursday

Today, the plan is to study about the new European Union (EU) regulation on data protection called the General Data Protection Regulation (GDPR) and take steps towards implementing it into our business operations.

According to the website of the European Union this new regulation is planned to take effect on 25th of May 2018. It warns that organizations that do not comply with its law are going to face heavy fines. To conduct business smoothly it is important to obey the law, protect customer data and respect their privacy.

Reviewing this new law will help us to know and decide what kind of changes we will need to make. And we already have started outlining additional data properties required to fulfill the new law.

Friday

From one of our Polish loan brokers, we have received a request asking if we could send them a notification message when loan applications brokered through them are disbursed to customers. This would help them for following up on the status of the applications they sent, deal with any kind of issues that might happen and also help them get more refined statistics of data.

This particular customer interacts with us through web API and implementing this functionality did not need the writing of a new set of code. The reason is because we already have a generic message sending interface, for which I only had to assign the appropriate data to the properties of this interface and be done with the task.

Preparing some services that are aimed at doing generic tasks for example, sending messages, writing log information and database connection, in a loosely coupled manner enables code reusability and helps for a code base to scale easily with less friction. This particular task is one example of such.

Weekly Analysis

This week was comprised of small independent tasks that took me not more than one day to complete. Most of the tasks were dealing with existing code to add more functionality. Starting to learn about the new General Data Protection Regulation (GDPR) was also one of the interesting and important tasks of this week. In the coming week I plan to read more about this regulation to get more knowledge and understanding of it.

I also reviewed security roles policy for the two new employees joining us on Monday next week. We have a variety of security roles and must be assigned to new employees based on the team they are going to work and their position. I always take extra care while assigning active directory roles policy.

Adding new SMS templates and SMS sending functionality for those customers who do not use email was one of the small tasks of this week. This task was an example of how we always work to satisfy customer needs and how fast we respond once we identify or come across new customer requirements.

3.5 Week 5 (19.03.2018 - 23.03.2018)

Monday

This morning our batch jobs for delivering invoices, reminder letters, termination letters and warning letters failed, based on the fact they were working the day before, I suspected the problem started a little before I arrived to office. In such cases we refer the log files of each operation to know the reason and act upon. Based on the information from the log files I found out that the remote file server of our partner mail delivery company was unreachable.

And shortly after this we received information from the company that they are facing some technical problems with their file server and it could take three to four days to fix get and get live again. Although this was an unexpected problem, it must be priority and taken care of fast so that invoices can reach to our customers at the right time.

As a temporary fix, I updated the delivery operations code to send the invoices only through email attachments and commented out the streaming of the invoices to the currently not working file server of our partner mail delivery company. In addition, I also set the delivery status of these invoices to not delivered. This status helps our delivery operations to send the invoices again by mail once the partner file server is up and running.

Tuesday

We store credit application indicators for some of the loan applications that come through brokers. These indicators help us in the decision making of a loan approval process. A credit application indicator object contains a code source, code and its respective value, for a specific credit application and it is these properties our loan approval process takes into account during the decision process.

For today my plan is to fix a small bug found in the storage of new credit application indicators. We found that some of these indicators were stored more than once for the same credit application. This task is going to involve, first, perform a new test to identify the bug causing these duplicates and fix it. Then, I plan to create and run an SQL server query that returns a result set of these indicators which contains duplicate rows for the same credit application and delete them, after which only unique rows will remain in the database table.

Through the test I found out the indicators were being stored without a check if they already exist. This was the reason for the duplicates. I updated the code in a way, first it checks if the to be stored indicator already exists using code, code source and credit application. If the indicator already exists then the new value is checked against the existing one and is only updated if they have different values. I also added code to modify the “update time” property of an indicator to track the time of last update. Otherwise, if an indicator does not exist it gets stored as a new one.

I constructed the query for removing the duplicate rows by utilizing the Transact-SQL function ROW_NUMBER as a subquery inside a temporary named result set known as common table expression (CTE) that returns the count of rows containing the same values for a given set of columns. More about ROW_NUMBER can be found online from the official documentation of Microsoft about Transact-SQL.

Wednesday

As a continuation towards adding additional invoice delivery methods, today my plan is to implement an additional email delivery method to the already existing mail delivery method, for sending invoices to the invoices receivers of our invoices funding business operation.

Since the method for sending email is already in place, I only had to get ready the email template for invoice funding invoices and prepare parameters set for filling the dynamic fields of the template.

In this particular case of sending email to invoice receivers, I added an additional functionality that sends a notification email about a sent invoice and its details. The reason for adding the notification email is that in case of invoice receivers our CRM does not store the email activities, thus I found it necessary to add this additional method to send notification about the sent email.

Thursday

The content of the office 365 for business we use in our company is growing at a fast rate and is requesting for more storage space. In a general sense, although content can be kept indefinitely, it happens that a big chunk of it can become obsolete and no longer needed based on factors such as time and purpose. Thus today, my task is to create several retention policies that are going to be applied to the different data folders of our office 365 for business so that content we consider is important can be retained and that we no longer need or is irrelevant anymore can be get rid of.

A retention policy is a documented policy about the data governance of an organization. It specifies how long an information must be retained and when it can be disposed when it is no longer needed, based on factors such as the organization's regulatory compliance needs, content relevance and area of business.

Creating retention policies in office 365 is easy and is feature rich. For current requirements, I have created six retention policies for managing the content in our office 365. Some of these retention policies are set to retain important documents of customers for

indefinite period of time. They also include policies that retain content for a definite period of time and then delete it permanently thereafter. One of these policies is also set to find contents with specific keywords, which usually is test documents and emails produced during software development, retain them for few days and then delete them permanently.

Retention policies can be applied to different users or specific locations of an organization. These retention policies will reduce the use of unnecessary storage usage, make available only relevant data to our customer service and make our company comply with the financial industry regulations.

Friday

As a continuation to yesterday's task of creating retention policies to clean up content in our office 365 for business, today's plan is to complete a somewhat similar assignment of creating a data cleaning batch operation that removes no longer needed pdf attachments from emails that are older than a given period of time.

I started first by creating the SQL query that fetches all email attachments that are older than one year. Then foreach found attachment, I added a code to delete the attachment and then remove its reference values from its related email entity. The usual number of attachments we send per email vary between one to three attachments, and this batch operation is expected to significantly reduce the size of our database.

I performed several tests to ensure it is behaving as expected and it is deleting the attachments we want to be removed, then I added it to our batch jobs list to run daily in the morning and set it to perform one thousand delete operations per day.

Weekly analysis

This week was one the busy weeks filled with high priority tasks that touched the areas of transact-SQL, office 365 business administration and batch operations. Besides this, I also completed some non-high priority tasks such implementing email sending functionalities and helping out customer service with some technical issues they faced.

For doing the task of deleting duplicate records, I learned and made heavy use of the transact-SQL functions ROW_NUMBER, Over and Partition by. I used the ROW_NUMBER function to assign temporary incremental values to rows, so that it sets duplicate rows to have a value greater than one, which then makes deleting of duplicates easy. Itzik

Ben-Gan (2016, Chapter 7) defines ROW_NUMBER as a function that assigns temporary incremental and sequential integers to the results of a query ordered by a set of columns that makes a row unique. Thus, I used this functions to construct the query for the task of deleting the duplicate records we had in one of our entities.

Another important thing I learned during this week was about retention policies and what they are meant for. I could say my prior knowledge on retention policies was very limited. Microsoft has a comprehensive and easy to understand online documentation about retention policies. Since retention policies also involve the deletion of data, the time limit they are set to retain a certain data and where they are applied needs careful and detailed planning. Otherwise important data can get affected by policies that delete data.

3.6 Week 6 (26.03.2018 - 30.03.2018)

Monday

Today's task is to create an Application Programming Interface (API) for one of our partner brokers. The purpose of the API is to return a list of loan allocation offers for a credit application brokered through them. We already have various APIs in place that expose different kinds of credit application services, and this task is a new addition to them.

I started by writing a service method that accepts the identification of a particular credit application, sent from an authenticated broker and returns the list of loan allocation offers for the application if they exist. I also added exception handling to take care of failures in case of data or operational errors.

Testing was done by using some applications that were brokered through the broker this API was being created for and some that were from others, to see how it behaves. After thorough testing I published the service to our development server so that I can start writing the API and consume this service for testing before publishing it to production.

I wrote this API in a way first it authenticates the request by checking the credentials sent in the posted data, then it proceeds to check that the posted data has a valid model state

and is sent using the expected content type, we usually accept either JSON or XML content types by default. It is after this checks the API service gets consumed and returns the response with the corresponding status code based on the data received from the service.

I made several tests of the end point using Postman, which I could say is one of the most popular and widely used API development environments, and finally published it to production.

Tuesday

The same credit application broker I created an API for yesterday also requested for a new API that return a list of cancelled loan allocations for an application brokered through them. I confirmed this with my senior and proceeded to implement it.

Since I created an almost similar API the day before this task was done quite faster. I copied the service that returns loan allocations and used it to create a new service that returns cancelled loan allocations by refactoring its code. I also used the same code for exception handling and made small changes to fit this new requirement.

This service was similarly published to test server so that I can use when testing the API that I was going to write. From our API side I wrote the code for the action that returns the cancelled loan allocations, made tests and published it to production.

For the API endpoints I created today and yesterday, I used the library called AutoMapper, to map the returned loan allocations to the corresponding view models of our APIs. This library default mapping strategy is to map between properties with similar names. In case there is a need to map between properties with different names, it also allows you to define your own custom mappings. I used it to map the loan allocations I got from the service to the view models of the corresponding APIs, with just one line of code, instead of the traditional way of creating a for loop to map each item with in a list.

AutoMapper helps save a lot of time, when mapping between entities with many data properties, by removing the need to explicitly write mappings between each property, which also helps have a cleaner code base. More can be read about AutoMapper from its official documentation found on GitHub at the url: <https://github.com/AutoMapper/AutoMapper>.

Wednesday

Today I received the task to create API notifications for broker applications that are verified, cancelled or rejected by our respective processes. As an attachment to the task, I also received a file that contains the documentation for making API requests to the loan broker endpoints.

In each of our processes that verify, reject or cancel credit applications, there already is a step that sends notifications to brokers if the application being processed is through a broker. Thus, in this task, I had to add a block of code that checks if the processed application belongs to this particular broker, and if it does, make an API post request using the format specified in the attached document.

I constructed the required data format, which was JSON and required http method was to use post. Then after making several tests using Postman, I published the new changes and notified the broker to monitor and let us know in case of any issues.

Thursday

To increase our reach to more customers, I have already created two new dashboards into our CRM. These dashboards list customers our algorithm fetched as being potential for marketing and cold calls. Towards this step of increasing our marketing activities, it has been decided by my seniors to use MailChimp in our marketing activities.

In order to get this started, we have to integrate MailChimp with our marketing list batch operation. Thus today, my task is to get information about the services MailChimp provides, the kinds of subscriptions they have and guide on how to use their APIs for automating marketing emails.

I have read the documentation from the official website of MailChimp, which I found it straight forward, detailed and easy to understand. I also made a test account and proceeded to make some test templates and design the schema for customer marketing lists. I believe I have got enough information to get started and next week with the help of my senior, we will start writing the interface code for MailChimp.

Friday

Today is public holiday.

Weekly analysis

This week I spent most of it doing tasks related to creating API endpoints and the backend services to support them. Although I was already familiar with writing the services and controller for the APIs, my knowledge of the library AutoMapper was basic.

The tasks of this week involved a lot of data mapping for which I used AutoMapper to do most of the heavy lifting. AutoMapper mapping is based on the convention of mapping between properties of objects having the same name. But I also learned that it can also be customized to map between object properties with different names and used it in one of the tasks. For organizations like us, that are data driven and having relatively big databases, AutoMapper can be handy for automatically mapping of data between datasets that have hundreds of columns, which saves a lot of time and reduces human error.

Another important thing I learned this week was about MailChimp: a marketing automation platform. I learned how to create different groups of customers, how to use existing or create new email marketing templates and how to use their API to populate customers or subscribers list and other dynamic fields.

3.7 Week 7 (02.04.2018 - 06.04.2018)

Monday

Today is public holiday.

Tuesday

I spent last Thursday reading the documentation of MailChimp, through which I was able to have an insight on its capabilities and how it can be consumed. I also practiced creating of some sample marketing email templates and subscriber's schema, in order to get a firsthand experience of MailChimp tools.

Today my plan is to implement a MailChimp interface project. This interface is going to container the data models for our email marketing subscribers, dynamic fields of email templates and also methods to interact with the APIs of MailChimp.

To begin with, I received a requirements list from my senior, explaining the data we want to store regarding our email marketing subscribers, the operations of the method for creating and updating subscribers and also the configuration settings for interacting with MailChimp APIs.

First, I created the object structure(Model) for our subscribers, then wrote the code for the methods to post subscribers data in batch. According to MailChimp, the maximum number of subscribers we can post at a time cannot be more than five hundred, although our the batch job I am going to write is going to handle grouping the subscribers by number, I added an extra check for safety.

At the end of the day, I created the corresponding schema in our MailChimp list to which we are going to post subscribers in batch. The data types of these models is set to string, and I handled null properties to be converted to empty, by using C-Sharp's helper method `NullToEmpty`.

Wednesday

Today my plan is to continue yesterday's task of creating the methods that are going to post the list of our subscribers to MailChimp for marketing purposes and finalize it by creating the batch operation that is going to continuously add or update subscribers on a daily basis.

Before I started implementing the batch operation for MailChimp, I tested the method which I created yesterday for posting subscribers to MailChimp, so that I can call it safely from the batch operation I am going to code.

I wrote the code for the batch operation in a way to post couple of tens of thousands of subscribers on its first day of operation, then from that day onwards to post five thousand daily in the morning. According to MailChimp, a single post to their server can accept a batch containing a maximum of five hundred entities. But in our case, I set it group the list of subscribers into smaller subgroups each of them having two hundred subscribers, because we decided not to make the server busy by sending batches having the maximum number of subscribers, which as I mentioned above is five hundred.

After I completed writing the code and started testing the batch operation, I found out that the MailChimp server often run out of time, when processing multiple batches, we posted

iteratively. Thus, I handled this case by adding a flag that denotes if a batch was successfully posted to the server, and if not, it made the posting process to delay for a couple of seconds in order to give some time for the server to process what is already in its queue and try posting again.

Thursday

As a follow up to our MailChimp implementation, today I will spend some time monitoring the subscriber's data uploaded to the MailChimp server.

In addition, I received a new requirement to add new data properties to the Subscribers schema and also add filtering to exclude subscribers with email addresses belonging to certain email domains we are required to exclude.

There was not much coding involved today, except small changes to the subscriber's model from our side, and also to the schema in MailChimp. Then I also adjusted our subscribers fetching query to exclude subscribers having emails with the domains we are required to exclude.

Friday

Today my goal is to install a payment processing software that can connect and work smoothly with any bank that uses an EBICS (Electronic Banking Internet Communication Standard) server. We have a payment processing partner company that uses an EBICS server, and the ultimate goal is to change the current setting of single user installation of this program into multi user installation.

The need for the multi user installation aroused from the fact that the current installation resides locally in one of our customer service member, and we want other members to have access whenever the current member is absent from work.

First, I installed the program along with the workspace, in a shared network location, with read and write permissions set for users responsible for using this program. The program also allows to create additional users once a user is logged in with administrator privileges. I took advantage of this feature and proceeded to create additional user accounts and completed the task.

Weekly analysis

I spent a fair share of this week learning about MailChimp. As I have mentioned MailChimp is a marketing automation tool and can be used to automate a lot of marketing activities such as creating marketing templates, forms and lists that contains subscribers' data and tools to manage them. I was able to read its documentation and gained a fair amount of knowledge regarding its features and how to use its APIs. I also learned about EBICS, which could be useful for me in the future.

The rest of the week was spent implementing the interface and batch operation that has methods which interact with APIs of Mailchimp and handle scheduled running and fetching of subscribers from our database respectively.

3.8 Week 8 (09.04.2018 - 13.04.2018)

Monday

We already have an API that enables some of our business partners to send us important customer documents one at a time. Today's goal is to create a new API that will enable these partners to send customer documents in bulk by placing them in a zipped folder. Customer documents will be renamed and placed inside the zipped folder according to a convention we have agreed upon. This convention will also help me in creating the algorithm of the service for processing the zipped folder.

.NET Framework already has a rich class library for processing a zipped folder. In this particular case I made use of the methods `Open` and `ExtractToDirectory` from one of the classes of the library called `ZipFile`, to open and move the files inside the zipped folder to the target location. In addition, I used the directory and file management methods, to create and write directories and files.

After testing, I published it to our test server and proceeded with developing the front-end API endpoint. While creating this, I made use of an existing code that streams pdf files and refactored it to stream a zipped folder containing many files. Then finally, published it to production. We look forward on how big performance improvement this new API is going to bring to our business partners.

Tuesday

We have a new employee starting next week, who is going to join the customer service for the newly starting Swedish market. We have got a new computer for him, and today I am going to spend configuring the computer and also setup the different accounts that are necessary to get him start working.

I usually start configuring accounts for new users, first by creating an account in Active Directory (AD). This account is crucial for a new member to get started using the CRM and access shared network data directories. I can also control the access level, rights and security of user accounts from the active directory. So, for the new employee I created an account and a temporary password which he will be prompted to update it with his own during the first sign in attempt.

I continued by installing our company's Virtual Private Network(VPN) program and certificates that are going to enable the new employee to work remotely and access the CRM and other resources. Then I purchased Office 365 premium license to be installed in the new computer.

Wednesday

Today we received a request to submit reports of business activities for the anti-money laundering department of a partner company. These reports are related to loan activities and a manual about our loan operation procedure.

I generated most of the require reports using the respective batch operations responsible to generate each of them and adjusted their parameters, to get the reports according to the requirements from the partner company's department.

We use Secure File Transfer Protocol (SFTP) to share files, and I used the popular software FileZilla to send the generated reports.

Thursday

Some of our customers unsubscribe from the marketing emails we send them through MailChimp using the unsubscribe link found in the email. In such cases, MailChimp server takes care of the updating the customers status so that they will not receive marketing

emails from us until they subscribe back. But it also happens often that some of the customers rather than unsubscribing themselves, they contact our customer service and request if they could be removed from the email marketing list. For such requests, our customer service updates the customer subscription status to unsubscribed locally, then the IT department takes care of persisting this changes to MailChimp. With the rapid growth of our customer base, we found it crucial to automate the latter mentioned process.

Since this task is about MailChimp subscribers, I will add a new unsubscribe method inside the already existing batch operation for MailChimp. Adding it to the batch operation gives the advantage for it to run automatically and continuously based on the scheduled run time of the batch operation, which in turn will make our list of subscribers in MailChimp up-to-date.

Mailchimp has a comprehensive documentation, along with examples and playground, on how to consume its API's. And since this task is about making a partial update to a subscriber's data, the subscription status to be exact, I had to use the Patch http request method. I checked the MailChimp documentation on how to Patch subscribers' data and wrote the method in a way that Patch subscribers data in a bulk.

Friday

We have got two additional new employees starting next week. I have received two new laptops and I will spend the day configuring these laptops and creating accounts for the new employees in a similar way I prepared for the other user on Tuesday.

Weekly analysis

As I might have mentioned multiple times, we are continuously working towards entering the Swedish market in a shorter time. And as a matter of fact, we have three new employees starting next week. I spent some days of this week configuring new computers and preparing different kinds of user accounts for them.

One another important thing I learned this week was also about processing Zipped files. I did not have prior experience regarding how to process Zipped files and I always thought the learning curve is not easy. But through a task i had this week, that involved processing a zipped folder sent over to us through API, I was able to learn and create a service that opens a zipped folder and process its contents. I also experimented on how to identify subdirectories from files inside a zipped folder and extract files to a desired location.

Regarding our marketing activities, it was decided to continuously check our existing subscribers base in MailChimp and update their subscription information, for which I learned how to partially update data in MailChimp through their documentation on how to make Patch requests. In addition, I also learned about handling MailChimp server timeouts and fail outs while sending large chunks of data.

3.9 Week 9 (16.04.2018 - 20.04.2018)

Monday

Today's task is to update the email templates we use for German customers. The change will involve updating the textual content and also adding additional new dynamic fields that populate customer specific data into these templates. Moreover, I have to refactor our email generation method, to automatically add a new pdf attachment to the emails we send that are about loan collection, payment reminders and loan termination.

First, our customer service team for Germany made changes to the text content of these email templates. They also marked the locations of the dynamic fields within these templates. Some of the dynamic fields were new, for which I had to create new properly formatted parameter sets that populate them. Formatting of dynamic fields involves, for example, if one is a currency field, I use C-Sharp's data conversion and format specifier methods to generate the correct value along with the correct currency symbol of the target customer.

For the pdf attachment, I created a method that make use of the .Net framework's File class library to locate the correct file from our Template's directory and return its full path. The I used this method in the email generation method to add the required new attachment if an email is about loan collection, payment reminders or loan termination and is being sent to a German customer.

After testing the new changes, I also made a quick update to the layout of the updated templates by using the on-demand editor of the popular frontend developers' playground codepen.io and finally published to production.

Tuesday

The number of users accessing our API's is growing, and today the plan is to add authentication to some of them. Although, these API's have an identification property set to identify external users, we found it important to add an additional layer of authentication. For now, we have decided to authenticate our end point users using simple username and password. And in the near future we will update it to use token authentication for a better security of our endpoints.

I started by adding the value for username and password for the customers who we have allowed and agreed to use our API, into our configuration files. These customers are usually loan brokers, and most of the time they consume our services using API's. Then for the changes to take effect I rebuilt the solution and added additional validation for username and password in our validation method.

After publishing these new changes to production, I also informed the partners who consume these endpoints and exchanged with them the authentication credentials through a secure channel.

Wednesday

Our customer service makes a lot of loan offers, that might suit a customer based on factors such as income and credit history. After contacting a customer and getting the necessary information our customer service makes use of a loan decision service to get the amount of loan they can offer. The amount offered can either be accepted or declined by a customer, and if declined adjustments can be made based on agreement between us and the customer.

The process of getting initial loan decision and then making adjustments if customer does not agree to the initial amount, require several roundtrip operations among our server, database and the front-end CRM. With the increasing number of our customer base and the number of offers we make, our database gets busy at times. Today we plan to optimize this process.

So, along with my senior we refactored our loan decision service, in a way it gives the initial loan decision without persisting the data to database, by making use of temporary memory, and only persist the changes after an agreement is reached between us and a customer.

Although the refactoring did not involve huge change in the service's logic, the small change we made the decision process faster. We confirmed this by making several tests and also comments received from our customer service members.

Thursday

Today I will be preparing a new computer and user accounts for a new employee starting next Monday. After finishing creating the accounts, I will also move an SMS gateway from the current project it resides into our web API project. Customers use this SMS gateway to check the status of their application and other relevant information.

Creating the user account was done same way as I have mentioned multiple times in this diary. And also moving the SMS gateway project involved the importing it to our API project solution and reconfiguring some settings in order to be able to function along with other API endpoints. Then, finally I informed our partner that directs SMS inquiries into our service, to use the new API we created.

Friday

Today we will have a workshop on how to use the JavaScript library Knockout and integrate it with our existing Model View Controller (MVC) architecture. Knockout is a light-weight library that helps in creating rich and responsive user interfaces (UI) and works seamlessly with underlying data model.

The workshop was given by my senior and our company's COO. I could say that I learned the capabilities of Knockout and how it contributes in developing web applications in a shorter period of time. I look forward to get started using the Knockout library and start learning by doing.

Weekly analysis

Within this week, I did a variety of tasks and had a workshop on Friday. In the workshop I we were given some introduction about the JavaScript library Knockout and how it integrates well with the Model View Controller (MVC) or Model View View model (MVVM) architecture for developing web applications. This workshop was a good jumpstart for me and I will continue to study the library and experiment more with it.

Another task I did was also about adding authentication to some of our APIs. The authentication I did was a simple username and password for the time being. But the long-term loan is to make the authentication using web tokens specifically JSON Web Token (JWT). I did not have prior experience regarding JWT before and I started learning about it from their website's online documentation.

3.10 Week 10 (23.04.2018 - 27.04.2018)

Monday

For the purpose making data requests faster, I have received a new task from my senior to modify some indexes of an existing SQL database table which stores loan allocation data. As stated by Itzik Ben-Gan (2012, 22.), indexing is a way or mechanism used by a database server, to improve the database performance and speed up queries, by creating unique combination of rows and avoiding unnecessary scanning of all columns of a table.

Indexes for SQL server tables can be created in several ways. One can use the windows PowerShell to script and execute indexing commands. In my case, I used SQL Server Management Studio (SSMS) to modify the targeted indexes and execute them to persist the changes. In SSMS existing indexed can be modified, either by double clicking on the index, which gives a wizard to make changes. And the other way, is to right click on the index and then click the sub menu list item "Script Index as" which has options to create, drop and edit indexes.

I used the query editor window and opened the existing indexes with the menu option that drops the existing index and recreates it again with the same name and newly added columns if there are any. Thus, in the query editor I modified the indexes by adding the columns specified by my senior and saved the changes to do the execution later in the evening.

The reason that we decided to do the execution of recreating the indexes after I modified them was, because persisting the changes of the modified indexes takes time and can slow the database response. This was the reason we decided to do the execution, in the evening where there are less database operations when compared to working hours.

Finally, I successfully executed the queries of the modified indexes, in the evening after I arrived home from work, using a remote machine. We look forward to see improvements in our data services.

Tuesday

As part of our steps towards implementing the General Data Protection Regulation (GDPR), today my task is to implement a process that deletes customer data. The data subject rights section of the GDPR Key Changes states that, an individual has the right to ask, access be informed and delete his/her own data (GDPR Key Changes 2018). Thus, this is one of the steps towards fulfilling the requirements of the GDPR.

The process I am going to implement is going to completely remove the data the user required to be removed from our data storage. It is going to have several steps that perform delete operations on the various sections of a user's data if they exist. For example, a user might want to completely remove his/her data, and another one might only want to remove the data about his/her credit applications but keep the contact information.

Implementing this task was quite easy forward, that I needed to make conditional checks on what type of data the user has chosen to remove and then perform delete operation after the user has confirmed his/her decision.

Wednesday

Today I will be refactoring our server-side validator of bank account number for Swedish customers. The current implementation of the validator checks for the validity of account numbers in their local format. I will add new functionality to check that the submitted account number in IBAN (International Bank Account Numbers) format, has the correct number of digits based on the country it represents, and also perform a "sanity-test" to check its validity.

The steps on how to validate an IBAN can be found online on many banking websites and even on Wikipedia. The first steps involve converting the IBAN which is a combination of letters and numbers into an integer (IBAN VALIDATION - WHAT YOU NEED TO KNOW FOR GLOBAL MONEY MOVEMENT 2018).

I read the procedure of validating an IBAN from different sources, to make sure the algorithm they mentioned is the same across all of them. Then according to the standard ISO 13616:2007, I added logic to change the country code part, which is non-numeric, into numeric values. Then after making sure the IBAN is changed to an integer value, first four leftmost digits are cut and moved to the right position and finally the remainder of the number is checked by dividing it to the value 97. If the result is equal to one it means the IBAN is a valid IBAN otherwise it is an invalid account number.

I finalized today's task by making several tests using various types of account numbers.

Thursday

Today I will modify one batch operation that generates loan realization files for one partner company. Currently the batch operation generates csv and zipped files and saves them in a folder having name same as the partner company. Then these files are sent to the partner manually whenever necessary.

My plan for today is, after these files are generated, to save them in a folder and set the name of the folder to be the value of the date the files got generated, and finally the folder will be sent to a file server of the partner company through SFTP (Secure File Transfer Protocol). I have already received the configuration parameters for getting connected to their server.

For creating the directory and then moving the files to that directory, I used the file and directory management classes of the System.IO namespace. Andrew Troelsen and Philip Japikse (2015, 749), mention that, "Many of the types within the System.IO namespace focus on the programmatic manipulation of physical directories and files."

Friday

For today, my plans are to create an account for a new employee joining our invoice funding team and also refactor the email sending method to include personal sender signatures when required.

For creating the account, I am going to follow the steps which I have mentioned several times in this document. The only thing I am going to add is, give additional access privilege for the new employee, which I use the Active Directory management to do this. Besides I will purchase Office 365 business premium license and order Skype for Business account.

I refactored our email sending method, by adding two new optional parameters of Boolean type, that indicate if the email should use a personal signature and if the sender email should also be personal. If these parameters are set to true, the method will dynamically

add the customer service member's personal email address and signature at the end of the email body, otherwise it will use the default signature of our company.

Weekly analysis

Through the tasks I did this week I was able to learn some new things and also refresh some of my previous knowledge regarding how to create better database indexes and writing robust regular expressions for text matching and filtering.

Jason Strate and Grant Fritchey, explain that, indexes are a means to get data from a database in the fastest possible way (2015, 1). With the help of this book and various materials from the internet, I was able to get more deeper understanding of clustered, non-clustered and heap indexes are when to use them. This helped me to assess and identify that creating a non-clustered index was the better option to implement that task I had on creating indexes.

3.11 Week 11 (30.04.2018 - 04.05.2018)

Monday

We are making different kinds of preparations for our entry to the Swedish market. These preparations include things for example, adding Swedish language to our public, borrower and investor sites and hiring new employees to join the Swedish customer service.

Another important task I am going to implement today as part of this preparation is to create different kinds of reports for Swedish customers. These reports include documents such as invoices, reminder letters, collection letters, warning letters and termination letters.

For creating the reports layout, I will use the report designer of Microsoft Visual Studio, which generates an XML-Structured format of the report called the Report Definition Language Client Side (RDLC).

While creating the reports, one of the most important things I pay special attention is the location of the address window in the reports that are customer specific. Properly placing the customer's full name and address ensures they appear properly through the envelope

window which makes it easy for a mail carrier company to deliver it to the right customer and at the right time.

Tuesday

Today I will continue yesterday's task of creating reports for Swedish customers.

Wednesday

One of the new employees for Swedish customers service is helping us with translation of documents and email templates to Swedish language. To make the process faster, today I will teach him how to create email templates in CRM so that he can do the translation, insert the content and do the formatting of the templates by himself instead of sending translated text to me.

Since we use HTML and CSS to format our email templates, I showed him how he can use the w3schools.com online editor to design templates and check the result instantly. I started by showing him one demo on the website and we also visited other alternative online editors such as CodePen and Plunker, so that he can pick any that he feels more comfortable and easy to use.

This is going to save us a lot of time, but since he is just a beginner with HTML and CSS I will help him with technical difficulties he might face and also check out the templates he creates, before putting them to production.

Thursday

Today I will be setting up a new room in our office to be used by four new employees starting next week. Since it is a new room, which was locked before, I will place newly purchased four 27-inch computer monitors and add wired internet connection to the room by using cables from our router.

Friday

Yesterday I spent it setting up the office room for the new employees that are going to start working next week. And today I will spend it on preparing accounts for them as well as install the necessary programs into their computers.

Weekly analysis

This week was spent on doing tasks, that I already know or I have done similar task before. Although I could say there were not much new things to learn, I was able to revisit and refresh my knowledge of creating reports in visual studio. I read about how I can include fonts that has diacritic letters in my production build, should they seem important to include to. This is a typical case when creating reports for customers that are from Eastern Europe, where diacritic letters are often found in their names as well as their addresses.

With the current fast paced innovation in software development, I have also learned it is crucial to stay relevant and updated. This includes not only learning new things, but also updating what I already know with better alternatives. One example is the .Net pdf library iTextSharp, which has features that are not available in Visual Studio report designer.

4 Discussion

Looking back to the day I started working as a backend developer in Fellow Finance and reviewing my growth in terms of problem solving capabilities, technical abilities, communication in and outside of my organization, handling emergency situations and social interaction, I could say it was a huge leap forward, which has resulted for me becoming a senior developer.

When I started working in Fellow Finance, although I was able to write code, it was not enough to just get started and start developing services. In order to get started it was crucial for me to learn about credit applications, loans, customers, payment processing and many others financial terms so that I can understand the business processes and start writing meaningful code.

Majority of the tasks I got assigned to complete required at least some knowledge of what credit applications, credit scoring, loans and credit scoring are. As a result, I was able to gain a lot of knowledge by referring materials from the internet and also asking for help from my seniors when necessary. The effect of learning about these terms not only improved my problem-solving abilities, but it also improved my communication skills with customer service members, whenever any issue regarding credit applications, loans or customers arises.

Regarding software development, the diversity of the tasks ranging from CRM dashboards development using XAML and Silverlight plugins, writing complex SQL queries and writing different kinds of services, presented me with both challenges and at the same time opportunities to learn and deepen my knowledge. I also came to realize the effectiveness of learning by doing in understanding the finer details of the task at hand. I experiment a lot with SQL and the C Sharp language using online editors and keep note of theories and code snippets for future use.

I often use the development methodology of build, measure and learn stated in the book *THE LEAN STARTUP* by *Eric Ries*. Eric Ries (2011, 126) stated that, "If you are building the wrong thing, optimizing the product or its marketing will not yield significant results". I usually apply this methodology to keep myself in check that what I do is going to make a change and yield a positive result. From this book I have also learned whenever building anything, it is a proven way to start developing the minimum viable product (MVP), which is the working version of the product that is built with minimum effort and in shorter time.

Then once the its results are checked and prove to bring value, more feature and functionalities can be added.

As I mentioned in one of the above paragraphs, from the challenges I faced in completing the tasks assigned to me, I was able to grasp few lessons regarding approaches of solving problems. First thing, I learned to keep record of important information such as code snippets of C Sharp and SQL queries, list of important reference books, websites and articles. Another important method I use when coding is, to first write the code that takes shortest amount of time and effort, then optimize it with time once proven yielding the desired results.

During the writing of this learning diary, the types of the tasks I did were related to developing different kinds of services, CRM, generating reports and creating complex SQL queries. Each type of task presented its own challenges, which I was able to solve and learn how to tackle such kinds of tasks in the future. For example, one of the tasks were about generating reports that include diacritic letters, which I learned to use a pf library called iTextSharp to generate the reports by embedding the fonts that has these diacritic letters. In this task I learned that the default Visual Studio report designed does not provide functionality to embed special characters and I also learned that sometimes third-party libraries are very useful in solving difficult tasks.

Another important thing I learned during the writing of this thesis was implementing several tasks related to creating batch operations that perform invoice generation, invoice printing, investor reporting and many other tasks. I learned how batch jobs are good choice for routine tasks and how to create them. Creating complex SQL server queries that perform filtering and subqueries was also one of the new things I learned.

I was excited to find out that the diversity of the tasks I do is so wide. When only at work, I was only paying attention to do my tasks. But, when I started writing the thesis and review my weekly diary entries, I feel fascinated and satisfied that I am able to participate and contribute to the backend, CRM, API and database of our company's platform. For the future, I plan to use these entries as a reference and improve upon them.

References

Andrew Troelsen, Philip Japiske. 2015. C# 6 and the .NET 4.6 Framework. 7th edition. Apress.

Eric Ries. 2011. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. 1st Edition. Crown Publishing Group.

Gary McLean Hall. 2010. Pro WPF and Silverlight MVVM: Effective Application Development with Model-View-ViewModel. 1st Edition. Apress.

GDPR Portal: Site Overview. URL: <https://www.eugdpr.org/eugdpr.org.html>. Accessed March 2018.

Greg Shields. Windows Server 2012 R2 (70-410) Administer Active Directory. URL: <https://app.pluralsight.com/library/courses/windows-server-2012-70-410-active-directory/table-of-contents?aid=7010a000002BWqGAAW>. Accessed January 2018.

How to: Sign XML Documents with Digital Signatures. URL: <https://docs.microsoft.com/en-us/dotnet/standard/security/how-to-sign-xml-documents-with-digital-signatures>. Accessed 26 February 2018.

Itzik Ben-Gan. 2012. Microsoft SQL Server 2012 T-SQL Fundamentals. 1st Edition. Microsoft Press.

Itzik Ben-Gan. 2016. T-SQL Fundamentals. 3rd Edition. Microsoft Press.

Jeffrey S. Beasley, Piyasat Nilkaew. 2012. Networking Essentials. 3rd Edition. Pearson Education, Inc.

Joe Booth. 2017. Regular Expressions Succinctly. CreateSpace Independent Publishing Platform.

Niel M. Bornstein. 2003. .NET and XML. 1st Edition. O'Reilly Media.